

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторної роботи
за темою «Ознайомлення з основними командами мови SQL, що
забезпечують маніпулювання даними на прикладі СУБД Microsoft
Access»

для студентів спеціальностей
035 «Філологія»
121 «Інженерія програмного забезпечення»
122 «Комп'ютерні науки»
126 «Інформаційні системи та технології»

Харків 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторної роботи
за темою «Ознайомлення з основними командами мови SQL, що
забезпечують маніпулювання даними на прикладі СУБД Microsoft
Access»

для студентів спеціальностей
035 «Філологія»
121 «Інженерія програмного забезпечення»
122 «Комп'ютерні науки»
126 «Інформаційні системи та технології»

Затверджено
Редакційно-видавничою
радою університету,
протокол № 2 від 25.06.2020

Харків
НТУ «ХПІ»
2020

Методичні вказівки до виконання лабораторної роботи за темою «Ознайомлення з основними командами мови SQL, що забезпечують маніпулювання даними на прикладі СУБД Microsoft Access» для студентів спеціальностей 035 «Філологія», 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки» та 126 «Інформаційні системи та технології» / уклад. Д. Л. Орловський, Н. В. Борисова, А. М. Копп. – Харків: НТУ «ХПІ», 2020. – 58 с.

Укладачі: Д. Л. Орловський

Н. В. Борисова

А. М. Копп

Рецензент Гринченко М.А.

Кафедра програмної інженерії та інформаційних технологій
управління

Кафедра інтелектуальних комп'ютерних систем

ЗМІСТ

ВСТУП.....	4
1 ТЕОРЕТИЧНІ ВІДОМОСТІ.....	5
2 ПІДГОТОВКА ДО РОБОТИ.....	13
3 ВИКОНАННЯ РОБОТИ.....	14
4 ВИМОГИ ДО ЗВІТУ.....	54
5 ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ.....	55
СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ.....	57

ВСТУП

Досвід застосування ЕОМ для побудови прикладних систем обробки даних показує, що найефективнішим інструментом тут не є універсальні мови програмування, а спеціалізовані мови для створення систем управління даними. Такі засоби зазвичай включаються до складу систем управління базами даних (СУБД), але вони можуть існувати і окремо. СУБД дають користувачам можливість здійснювати безпосереднє управління даними, а програмістам – швидко розробляти програмні засоби для їх обробки.

У методичних вказівках розглядаються питання створення бази даних, роботи з даними і вивчення основ програмування в середовищі СУБД Microsoft Access. СУБД Microsoft Access отримала в даний час дуже широке поширення. Це в першу чергу пов'язано з тим, що дана СУБД входить до складу інтегрованого пакета програм Microsoft Office. Поряд з текстовим процесором Word і електронною таблицею Excel СУБД Access, широко використовується для вирішення різних завдань, пов'язаних з автоматизацією діловодства та документообігу. Наявність досить простого інтерфейсу, розвиненої системи допомоги, а також різних вбудованих засобів (так званих майстрів), які полегшують процес навчання користувача, роблять СУБД Access дуже привабливою, особливо для користувачів, які не мають великого досвіду роботи з системами управління базами даних.

Виконання комплексу робіт обумовлює розгляд цілого ряду питань – від створення бази даних до розробки інтерфейсу користувача. В ході виконання комплексу робіт студенти матимуть можливість ознайомитися з різними засобами автоматизації завдань, пов'язаних зі зберіганням і обробкою даних. До таких засобів відносяться засоби побудови екранних форм, запитів, звітів і т.д. Дана лабораторна робота присвячена ознайомленню з основними командами мови SQL, що забезпечують маніпулювання даними. При виконанні лабораторних робіт передбачається використання СУБД Microsoft Access версії 2007 і вище.

Продуктивне освоєння можливостей СУБД Microsoft Access можливе тільки при практичній опрацюванні матеріалу на комп'ютері.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

SQL – це мова програмування для роботи з наборами фактів і зв'язками між ними. У програмах керування реляційними базами даних, як-от Microsoft Office Access, мова SQL використовується для роботи з даними. На відміну від багатьох мов програмування, мова SQL дуже проста. Подібно до інших мов програмування, SQL – це міжнародний стандарт, визнаний такими організаціями зі стандартизації, як ISO й ANSI.

Мовою SQL описуються набори даних, які можуть допомогти відповісти на запитання. Використовуючи SQL, потрібно застосовувати правильний синтаксис. Синтаксис – це набір правил, які забезпечують правильне поєднання елементів мови. Синтаксис SQL базується на синтаксисі англійської мови та має багато спільних елементів із синтаксисом Visual Basic for Applications (VBA).

Наприклад, ось як може виглядати проста інструкція SQL для отримання списку прізвищ з іменем Mary:

```
SELECT Last_Name  
FROM Contacts  
WHERE First_Name = 'Mary';
```

Інструкція SELECT використовується для опису набору даних на мові SQL. Інструкція SELECT містить повний опис набору даних, які потрібно отримати з бази даних, зокрема:

- таблиці, які містять дані;
- зв'язки між даними з різних джерел;
- поля або обчислення, на основі яких створюються дані;
- умови, яким мають відповідати дані, що включаються в результат;
- доцільність і спосіб сортування результатів.

Інструкція SQL складається з речень. Кожне речення в інструкції SQL виконує свою функцію. Деякі речення в інструкції SELECT є обов'язковими. У таблиці нижче наведено найпоширеніші речення SQL (таблиця 3.1).

Таблиця 3.1 – Найпоширеніші речення SQL

Речення SQL	Призначення	Обов'язкове
SELECT	Перелічує поля, які містять потрібні дані	Так
FROM	Перелічує таблиці, у яких містяться поля, зазначені в реченні SELECT	Так
WHERE	Визначає умови для поля, яким мають відповідати всі записи, щоб бути включеними до результатів	Ні
ORDER BY	Визначає спосіб сортування результатів	Ні
GROUP BY	В інструкції SQL, яка містить агрегатні функції, перелічує поля, не зведені в реченні SELECT	Лише за наявності таких полів
HAVING	В інструкції SQL, яка містить агрегатні функції, визначає умови, які застосовуються до полів, зведених в інструкції SELECT	Ні

Кожне речення SQL складається з термінів, які можна порівняти з частинами мови. У таблиці нижче наведено типи термінів SQL.

Таблиця 3.2 – Типи термінів SQL

Термін SQL	Частина мови	Визначення	Приклад
ідентифікатор	іменник	Ім'я, яке використовується для визначення об'єкта бази даних, наприклад імені поля	Клієнти.[Номер телефону]
оператор	дієслово або прислівник	Ключове слово, яке позначає або змінює дію	AS

Закінчення таблиці 3.2

Термін SQL	Частина мови	Визначення	Приклад
константа	іменник	Значення, яке не змінюється, наприклад число або NULL-значення	42
вираз	прикметник	Комбінація ідентифікаторів, операторів, констант і функцій, яка обчислюється до одного значення	>= Продукти.[Ціна за одиницю]

Загальний формат інструкції SQL:

```
SELECT field_1
FROM table_1
WHERE criterion_1;
```

```
SELECT [E-mail Address], Company
```

Це речення SELECT. Воно складається з оператора (SELECT), за яким слідують два ідентифікатори ([Адреса електронної пошти] та «Компанія»).

Якщо ідентифікатор містить пробіли або спеціальні символи (наприклад, «адреса електронної пошти»), його потрібно взяти в квадратні дужки.

У реченні SELECT не потрібно вказувати таблиці з полями чи умови, яким мають відповідати дані, що включаються в результат.

Речення SELECT завжди відображається перед реченням FROM в інструкції SELECT.

```
FROM Contacts
```

Це речення FROM. Воно складається з оператора (FROM), за яким слідує ідентифікатор («Контакти»).

У реченні FROM не перелічуються поля для вибору.

WHERE City="Seattle"

Це речення WHERE. Воно складається з оператора (WHERE), за яким слідує вираз (Місто="Сіетл").

За допомогою речень SELECT, FROM і WHERE можна виконати низку дій, використовуючи мову SQL.

Як і Microsoft Excel, програма Access дає змогу сортувати результати запиту даних у табличному поданні. У запиті також можна вказати спосіб сортування результатів під час виконання запиту, використовуючи речення ORDER BY. Якщо ви використовуєте речення ORDER BY, воно має розташовуватися в кінці інструкції SQL.

Речення ORDER BY містить список полів, які потрібно використовувати для сортування, у порядку, який потрібно застосувати до операцій сортування.

Наприклад, припустимо, що результати спочатку потрібно сортувати за спаданням значень поля «Компанія», а потім (за наявності записів з однаковим значенням для поля «Компанія») – за зростанням значень поля «Адреса електронної пошти». Речення ORDER BY матиме приблизно такий вигляд:

ORDER BY Company DESC, [E-mail Address]

Іноді потрібно працювати зі зведеними даними, такими як обсяг загального продажу за місяць або найдорожчі товари на складі. Для цього в реченні SELECT застосуйте до поля агрегатну функцію. Наприклад, якщо за допомогою запиту потрібно відобразити кількість адрес електронної пошти для кожної компанії, речення SELECT може мати такий вигляд:

SELECT COUNT([E-mail Address]), Company

Можливість використання агрегатних функцій залежить від типу даних у полі або виразу, що використовується.

Коли ви використовуєте агрегатні функції, зазвичай також потрібно створити речення GROUP BY. У реченні GROUP BY перелічуються всі поля, до яких не застосовується агрегатна функція. Якщо агрегатні функції застосовуються до всіх полів у запиті, речення GROUP BY створювати не потрібно.

Речення GROUP BY слідує безпосередньо після речення WHERE або FROM, якщо немає речення WHERE. У реченні GROUP

BY поля перелічуються в тому самому порядку, що й в реченні SELECT.

Продовжимо попередній приклад. Якщо речення SELECT застосовує агрегатну функцію лише до поля [Адреса електронної пошти], речення GROUP BY матиме такий вигляд:

```
GROUP BY Company
```

Якщо потрібно вказати умови, щоб обмежити результати, але поле, до якого їх потрібно застосувати, використовується в агрегатній функції, не можна використовувати речення WHERE. Натомість скористайтесь реченням HAVING. Речення HAVING працює так само, як і речення WHERE, але використовується для сукупних даних.

Наприклад, уявімо, що ви використовуєте функцію AVG (яка обчислює середнє значення) з першим полем у реченні SELECT:

```
SELECT COUNT([E-mail Address]), Company
```

Якщо запит має обмежувати результати на основі значення цієї функції COUNT, у реченні WHERE не можна використовувати умови для цього поля. Натомість умову слід включити в речення HAVING. Наприклад, якщо запит має повертати рядки, лише коли з компанією пов'язано кілька адрес електронної пошти, речення HAVING може мати такий вигляд:

```
HAVING COUNT([E-mail Address])>1
```

Оператор UNION використовується, щоб одночасно переглянути всі дані, повернуті кількома схожими вибірконими запитамі, у вигляді об'єднаного набору.

Оператор UNION дає змогу об'єднати дві інструкції SELECT в одну. Об'єднувані інструкції SELECT мають містити однакову кількість полів виводу, розташованих в однаковому порядку, з однаковими або сумісними типами даних. Коли ви запускаєте запит, дані з кожного набору відповідних полів об'єднуються в одне поле виводу, тому результат запиту містить таку ж кількість полів, що й кожна з інструкцій SELECT.

Використовуючи оператор UNION, ви також можете вказати, чи результати запиту мають включати повторювані рядки (за наявності). Для цього використовується ключове слово ALL.

Базовий синтаксис SQL для запиту на об'єднання двох інструкцій SELECT має такий вигляд:

```
SELECT field_1  
FROM table_1  
UNION [ALL]  
SELECT field_a  
FROM table_a;
```

Припустімо, що у існують таблиці «Товари» та «Послуги». Обидві ці таблиці містять поля, у яких зазначається ім'я товару або послуги, ціна, наявність гарантії та гарантійних зобов'язань, а також інформація про ексклюзивність пропозиції. Незважаючи на те, що в таблиці «Товари» зберігається інформація про гарантії, а в таблиці «Послуги» – інформація про гарантійні зобов'язання, основна інформація в цих таблицях така сама (наявність гарантії якості для певного товару або послуги). Щоб поєднати чотири поля з двох таблиць, скористайтеся запитом на об'єднання. Наведемо зразок такого запиту:

```
SELECT name, price, warranty_available, exclusive_offer  
FROM Products  
UNION ALL  
SELECT name, price, guarantee_available, exclusive_offer  
FROM Services;
```

В SQL всі записи в таблицю вводяться за допомогою команди модифікації INSERT. Але слід пам'ятати, що ім'я таблиці, в яку відбувається вставка, повинно бути попередньо визначене, а кожне значення, що вставляється повинне співпадати з типом даних стовбця, в який воно вставляється. Наприклад, для додавання запису в таблицю викладачів TEACHERS, можна скористатись наступним виразом:

```
INSERT INTO TEACHERS  
VALUES (4006, "Петров", "Петро", "Петрович", #01/09/1999#);
```

Можна вказати стовбці, в які необхідно здійснити вставку значень. Наприклад:

```
INSERT INTO TEACHERS (TDATE, TFAM, TIMA)  
VALUES (#01/09/1999#, "Петров", "Петро");
```

Для полів, які не вказані в запиті автоматично встановлюються значення за замовчанням.

Є можливість за допомогою команди INSERT отримувати чи вибирати значення з однієї таблиці та поміщати їх в іншу разом з запитом.

Видалення рядків з таблиці можна здійснити командою модифікації DELETE. Треба враховувати, те, що команда може видаляти тільки цілі записи таблиці, а не індивідуальні значення деяких полів. Для видалення всього вмісту таблиці STUDENTS можливо виконати наступне:

```
DELETE FROM STUDENTS;
```

Для визначення рядків, які треба видалити за умови, застосовують предикат. Наприклад, для видалення інформації, що стосується студента Нагорного можна використати наступну команду:

```
DELETE FROM STUDENTS  
WHERE SNUM=3416;
```

В якості предикату використовують номер студентського квитка.

Можливість зміни всіх або деяких значень в таблиці реалізується за допомогою команди UPDATE. В ній вказується ім'я таблиці, яка використовується та слово SET, яке визначає зміну яка відбудеться для потрібного поля таблиці. Наприклад:

```
UPDATE PERFORMANCE  
SET MARK = 5;
```

Приведе до зміни в таблиці PERFORMANCE всіх оцінок на «5».

Для зміни єдиного значення можна застосовувати предикати.

Наприклад:

```
UPDATE PERFORMANCE  
SET MARK = 5;  
WHERE PNUM = 2003;
```

За допомогою UPDATE можна модифікувати дані з декількох полів SET взмозі визначити будь-яку кількість полів, відокремлених комами.

Модифікувати зразу кілька таблиць однією командою UPDATE не дозволено, тому не можна застосовувати назву таблиці через крапку в іменах полів

У рядку SET команди UPDATE можна застосовувати вирази, розташовуючи їх в списку для того поля, яке необхідно змінити. Наприклад:

```
UPDATE STUDENTS  
SET SCHOLARSHIP = SCHOLARSHIP * 2;
```

Команда збільшує значення стипендії в 2 рази.

До недоліків команди UPDATE можна віднести неможливість посилатися на таблицю, яка задієна в будь-якому підзапиті з команди модифікації. Наприклад, неможливо одною командою виконати таку дію, як модифікація оцінок для студентів, у яких оцінки нижче середньої. Для цього необхідно виконати один запит:

```
SELECT AVG (MARK)  
FROM PERFORMANCE;
```

а потім результат цього запиту застосувати для модифікації:

```
UPDATE PERFORMANCE  
SET MARK = MARK - 1  
WHERE MARK < 4.2;
```

2 ПІДГОТОВКА ДО РОБОТИ

1. Записати в окремий каталог на диску ПЕОМ (наприклад, ACCLAB на диску D) файл бази даних, створеної в результаті виконання лабораторних робіт 1-2.
2. Провести запуск СУБД Access.
3. Відкрити базу даних, створену в результаті виконання лабораторних робіт 1-2.

3 ВИКОНАННЯ РОБОТИ

I. Команда SELECT-SQL. Обробка даних за допомогою SELECT-запитів

Для ознайомлення з синтаксисом команди SELECT-SQL методами побудови SELECT-запитів пропонується реалізувати ряд наведених нижче запитів. Для реалізації запитів необхідно відкрити вкладку «Create» вікна бази даних та обрати «Query design».

Увага! Наведені нижче результати виконання SELECT-SQL запитів відповідають стану бази даних, яка була створена у першій лабораторній роботі.

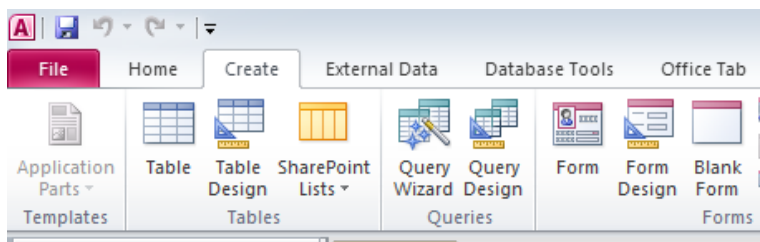


Рисунок 3.1 – Створення запиту

1. Вивести на екран список Продуктів, поставлених постачальником 1 (ПП Іванов І.І.) за договором 1.

Рішення:

- 1) відкрити вкладку «Create» (Рисунок 3.1);
- 2) у вікні «Queries» вибрати режим «Query Design»;
- 3) у вікні «Show table» вибрати по черзі і Add за допомогою кнопки «Add» таблиці «Contracts», «Supplied», «Suppliers». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язку між ними (рисунок 3.2);

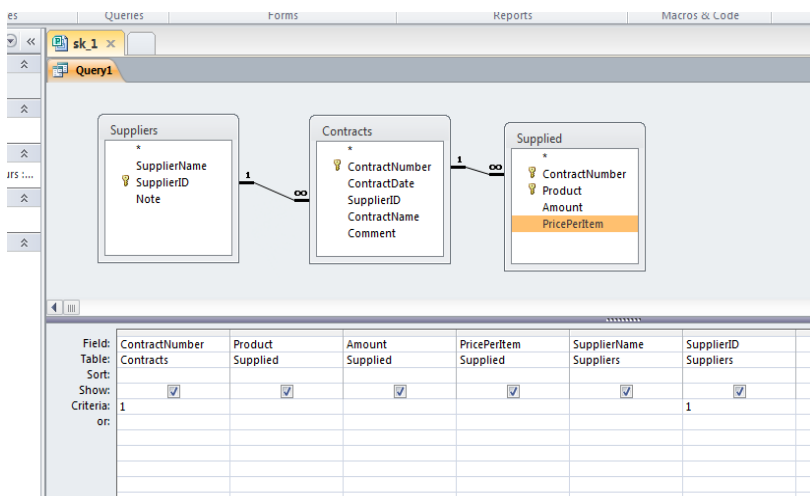


Рисунок 3.2 – Вікно конструктора запитів «Query designer»

4) сформувати список полів, які потраплять в результат запиту. Це такі поля (FirstNameТаблиці.FirstNameПоля): Contracts.ContractNumber, Supplied.Product, Supplied.Amount, Supplied.AmountPricePerItem, Suppliers.SupplierName, Suppliers.SupplierId (рисунок 3.2);

5) для полів «ContractNumber» і «SupplierId» встановити в Criteria значення 1 (рисунок 3.2);

6) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL View» (рисунок 3.3).

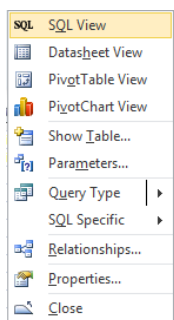


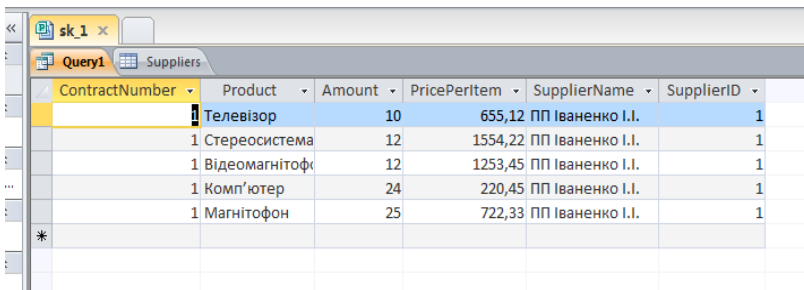
Рисунок 3.3 – Контекстне меню

У вікні буде приведений наступний текст:

```
SELECT      Contracts.ContractNumber,      Supplied.Product,  
Supplied.Amount,      Supplied.PricePerItem,      Suppliers.SupplierName,  
Suppliers.SupplierID  
FROM Suppliers INNER JOIN (Contracts INNER JOIN Supplied  
ON Contracts.ContractNumber = Supplied.ContractNumber) ON  
Suppliers.SupplierID = Contracts.SupplierID  
WHERE      (((Contracts.ContractNumber)=1)      AND  
((Suppliers.SupplierID)=1));
```

7) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

8) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок 3.3, 1.4).



ContractNumber	Product	Amount	PricePerItem	SupplierName	SupplierID
	Телевізор	10	655,12	ПП Іваненко І.І.	1
1	Стереосистема	12	1554,22	ПП Іваненко І.І.	1
1	Відеоманітоф	12	1253,45	ПП Іваненко І.І.	1
1	Комп'ютер	24	220,45	ПП Іваненко І.І.	1
1	Магнітофон	25	722,33	ПП Іваненко І.І.	1
*					

Рисунок 3.4 – Вигляд таблиці «DataSheet view»

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

9) закрити вікно запиту і зберегти запит з Ім'ям «Запит1» (без лапок).

2. Вивести на екран список Продуктів, поставлених постачальником 1 (ПП Іванов І.І.) в період з 05.09.1999 по 12.09.1999.

Рішення:

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «Queries design»;

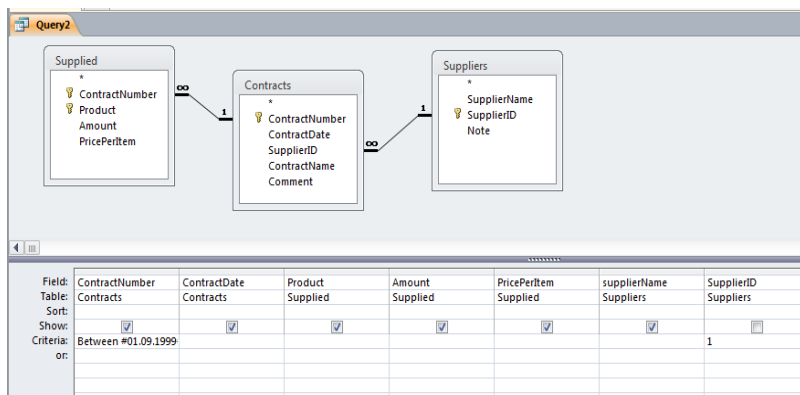


Рисунок 3.5 – Режим «Queries design»

3) у вікні «Show table» вибрати по черзі і Add за допомогою кнопки «Add» таблиці «Contracts», «Supplied», «Suppliers», вікно «Show table» закрити.

Внаслідок на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунок 3.5);

4) сформуванати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Supplied.ContractNumber, Contracts.ContractDate, Supplied.Product, Supplied.Amount, Supplied.AmountPricePerItem, Suppliers.SupplierName, Suppliers.SupplierId (рисунок 3.5). Для поля Suppliers.SupplierId відключити прапорець Show;

5) для поля «SupplierId» встановити в Criteria значення 1 (рисунок 3.5);

6) для поля «ContractDate» встановити в Criteria Expression: Between # 01.09.1999 # And # 12.09.1999 # (рисунок 3.5);

7) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
SELECT    Contracts.ContractNumber,    Contracts.ContractDate,
Supplied.Product,    Supplied.Amount,    Supplied.PricePerItem,
Suppliers.supplierName
```

```
FROM Suppliers INNER JOIN (Contracts INNER JOIN Supplied
ON Contracts.ContractNumber = Supplied.ContractNumber) ON
Suppliers.SupplierID = Contracts.SupplierID
```

WHERE (((Contracts.ContractNumber) Between #9/1/1999# And #9/12/1999#) AND ((Suppliers.SupplierID)=1));

8) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query view»;

9) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view»;

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

10) закрити вікно запиту і зберегти запит з ім'ям «Запит 2».

3. Вивести на екран список продуктів, поставлених в 9 місяці 1999 з висновком найменування постачальника і дати поставки.

Рішення:

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «SQL Designer»;

3) у вікні «Show table» вибрати по черзі і Add за допомогою кнопки «Add» таблиці «Contracts», «Supplied», «Suppliers». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунок 3.6);

4) сформувати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Supplied.ContractNumber, Contracts.ContractDate, Supplied.Product, Supplied.Amount, Supplied.AmountPricePerItem, Сума: [Amount] * [AmountPricePerItem], Suppliers.SupplierName, (рисунок 3.6). Поле «Сума» є обчислюваним і тому для нього FirstName таблиці не вказується;

5) для поля ContractDate встановити в Criteria Expression: Month ([ContractDate]) = 9 And Year ([ContractDate]) = 1999;

6) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
SELECT Contracts.ContractNumber, Contracts.ContractName,
Supplied.Product, Supplied.Amount, Supplied.PricePerItem,
Suppliers.SupplierName, [Amount]*[PricePerItem] AS Сума,
Contracts.ContractDate
```

```
FROM Suppliers INNER JOIN (Contracts INNER JOIN Supplied
ON Contracts.ContractNumber = Supplied.ContractNumber) ON
```

Suppliers.SupplierID = Contracts.SupplierID
 WHERE ((Month([ContractDate])=9 And
 Year([ContractDate])=1999));

7) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

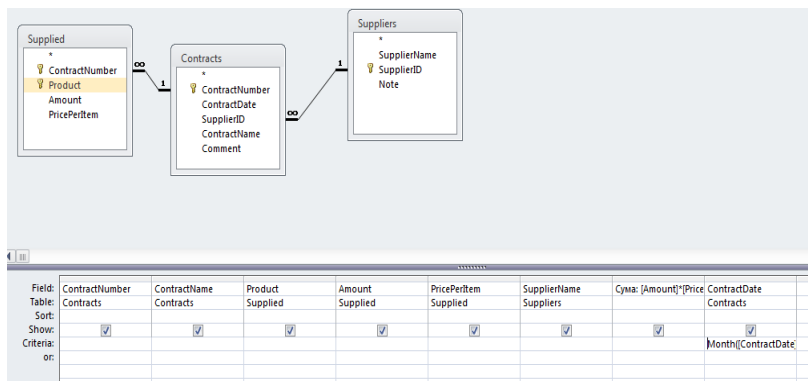


Рисунок 3.6 – Query designer

8) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок 3.7);

ContractNumber	ContractName	Product	Amount	PricePerItem	SupplierName	Сума	ContractDate
1	Договір № 1	Телевізор	10	655,12	ПП Іваненко І.І.	6551,19995117188	01.09.1999
1	Договір № 1	Стереосистема	12	1554,22	ПП Іваненко І.І.	18650,6396484375	01.09.1999
1	Договір № 1	Відеомагнітоф	12	1253,45	ПП Іваненко І.І.	15041,3994140625	01.09.1999
1	Договір № 1	Комп'ютер	24	220,45	ПП Іваненко І.І.	5290,79992675781	01.09.1999
1	Договір № 1	Магнітофон	25	722,33	ПП Іваненко І.І.	18058,2504272461	01.09.1999
2	Договір № 2	Магнітофон	5	511,43	ПП Іваненко І.І.	2557,14996337891	10.09.1999
2	Договір № 2	Відеомагнітоф	8	455,14	ПП Іваненко І.І.	3641,1201171875	10.09.1999
2	Договір № 2	Стереосистема	11	1453,18	ПП Іваненко І.І.	15984,9805908203	10.09.1999
2	Договір № 2	Комп'ютер	43	450,67	ПП Іваненко І.І.	19378,8105773926	10.09.1999
3	Договір № 3	Магнітофон	11	544	ПП Петренко П.П.	5984	10.09.1999
3	Договір № 3	Телевізор	52	545,32	ПП Петренко П.П.	28356,6403808594	10.09.1999
3	Договір № 3	Монитор	85	899,99	ПП Петренко П.П.	76499,1491699219	10.09.1999
4	Договір № 4	Магнітофон	22	323,19	ПП Петренко П.П.	7110,18005371094	23.09.1999
4	Договір № 4	Стереосистема	27	350,77	ПП Петренко П.П.	9470,78970336914	23.09.1999
4	Договір № 4	Принтер	41	330,55	ПП Петренко П.П.	13552,5494995117	23.09.1999
4	Договір № 4	Телевізор	56	990,56	ПП Петренко П.П.	55471,3598632813	23.09.1999
5	Договір № 5	Телевізор	14	590,23	ТОВ «Інтерфрут»	8263,2197265625	24.09.1999
5	Договір № 5	Відеомагнітоф	17	860,33	ТОВ «Інтерфрут»	14625,6102905273	24.09.1999
5	Договір № 5	Магнітофон	33	850,12	ТОВ «Інтерфрут»	28053,9598388672	24.09.1999
5	Договір № 5	Монитор	44	585,67	ТОВ «Інтерфрут»	25769,4792480469	24.09.1999

Рисунок 3.7 – DataSheet view

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

9) закрити вікно запиту і зберегти запит з ім'ям «Запит 3».

4. Вивести на екран список договорів (номер, дата), загальна кількість поставлених продуктів і загальну суму за кожним договором (розмір партії помножити на ціну за штуку і підсумувати за договором). Список повинен бути відсортований в порядку зростання номерів договорів.

Рішення:

1) натиснути кнопку «Create»;
2) у вікні «Create» вибрати режим «Query designer»;
3) у вікні «Show table» вибрати по черзі і Add за допомогою кнопки «Add» таблиці «Contracts» і «Supplied». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунок 3.8);

4) сформувати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Contracts.ContractNumber, Contracts.ContractDate, Supplied.Amount, ЗагальнаСума: Sum ([Amount] * [PricePerItem]) (рисунок 3.8). Поле «Загальна сума» є обчислюваним і тому для нього FirstName таблиці не вказується;

5) для поля ContractNumber встановити в рядку «Sort» режим «Asc»;

6) натиснути правою кнопкою миші в довільному місці таблиці зі списком полей запиту і включити режим «Totals». Для полів «ContractNumber» і «ContractDate» встановити режим «Group by», для поля «Amount» – «Sum», для поля «Загальна сума» – «Expression» (рисунок 3.8);

7) Переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
SELECT    Contracts.ContractNumber,    Contracts.ContractDate,
Supplied.Amount, Sum([Amount]*[PricePerItem]) AS ЗагальнаСума
FROM      Contracts    INNER JOIN    Supplied    ON
Contracts.ContractNumber = Supplied.ContractNumber
GROUP BY Contracts.ContractNumber, Contracts.ContractDate,
Supplied.Amount
ORDER BY Contracts.ContractNumber;
```

8) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

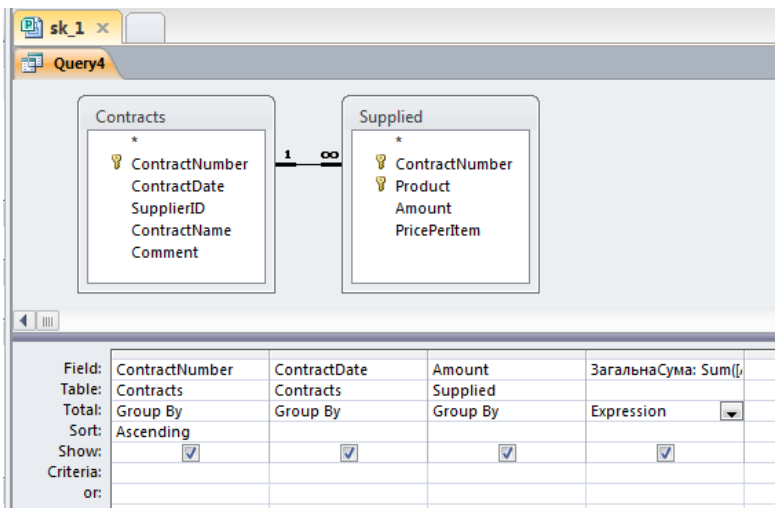


Рисунок 3.8 – Режим «Query designer»

9) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок 3.9).

ContractNumber	ContractDate	Amount	ЗагальнаСума
1	01.09.1999	10	6551,19995117188
1	01.09.1999	12	33692,0390625
1	01.09.1999	24	5290,79992675781
1	01.09.1999	25	18058,2504272461
2	10.09.1999	5	2557,14996337891
2	10.09.1999	8	3641,1201171875
2	10.09.1999	11	15984,9805908203
2	10.09.1999	43	19378,8105773926
3	10.09.1999	11	5984
3	10.09.1999	52	28356,6403808594
3	10.09.1999	85	76499,1491699219
4	23.09.1999	22	7110,18005371094
4	23.09.1999	27	9470,78970336914
4	23.09.1999	41	13552,5494995117
4	23.09.1999	56	55471,3598632813
5	24.09.1999	14	8263,2197265625
5	24.09.1999	17	14625,6102905273
5	24.09.1999	33	28053,9598388672
5	24.09.1999	44	25769,4792480469
6	01.10.1999	32	59207,6796875
6	01.10.1999	34	17712,3004150391
6	01.10.1999	51	41317,6512451172
7	02.10.1999	15	18518,4008789063
7	02.10.1999	22	8574,5

Рисунок 3.9 – Режим «DataSheet view»

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

10) закрити вікно запиту і зберегти запит з ім'ям «Запит4».

5. Вивести на екран список договорів (номер, дата) і загальну суму за кожним договором (розмір партії помножити на ціну за штуку і підсумувати за договором). Список повинен бути відсортований в порядку зростання загальних сум по кожному договору. Крім того, на список має бути накладено умова фільтрації, що складається у виключенні з результату запиту записів, для яких ContractNumber більше 3.

Рішення:

- 1) натиснути кнопку «Create»;
- 2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;

3) у вікні «Show table» вибрати по черзі і Add за допомогою кнопки «Add» таблиці «Contracts» і «Supplied». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунок 3.9);

4) сформуванати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Contracts.ContractNumber, Contracts.ContractDate, сума: Sum ([Amount] * [AmountPricePerItem]) (рисунок 3.9). Поле «Сума» є обчислюваним і тому для нього FirstName таблиці не вказується.

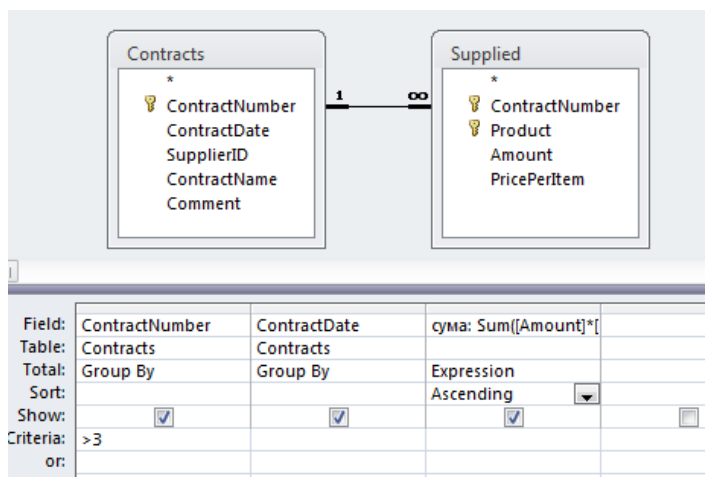


Рисунок 3.9 – Режим «Query designer»

5) для поля «Сума» встановити в рядку «Sort» режим «Asc»;

6) натиснути правою кнопкою миші в довільному місці таблиці зі списком полів запиту і включити режим «Totals». Для полів «ContractNumber» і «ContractDate» встановити режим «Group by», для поля «Сума» – «Expression» (рисунок 3.9);

7) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
SELECT  Contracts.ContractNumber,  Contracts.ContractDate,
Sum([Amount]*[AmountPricePerItem]) AS сума
FROM    Contracts  INNER JOIN  Supplied  ON
Contracts.ContractNumber = Supplied.ContractNumber
```

GROUP BY Contracts.ContractNumber, Contracts.ContractDate
HAVING (((Contracts.ContractNumber)>3))

ORDER BY Sum([Amount]*[AmountPricePerItem]); ORDER BY
Sum ([Amount] * [AmountPricePerItem]);

8) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

9) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок 3.10).

ContractNumber	ContractDate	сума
5	02.10.1999	297
6	24.09.1999	324
4	01.10.1999	351
4	23.09.1999	438

Рисунок 3.10 – Режим «DataSheet view»

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

10) закрити вікно запиту і зберегти запит з ім'ям «Запит5».

6. Вивести на екран відомості про найбільші за розміром комплекти з тих договорів із зазначенням постачальника, а також номери і дати договору.

Рішення:

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;

3) у вікні «Show table» вибрати по черзі і Add за допомогою кнопки «Add» таблиці «Contracts», «Supplied», «Suppliers». В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунок 3.11);

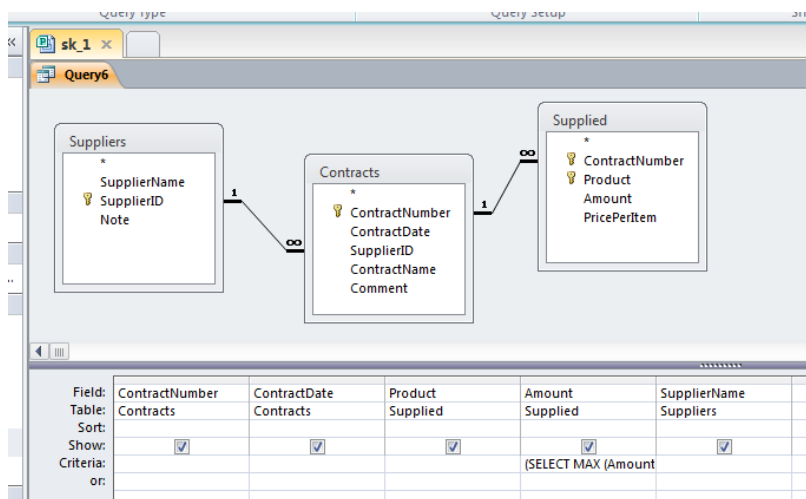


Рисунок 3.11 – Режим «Query designer»

4) сформувати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Contracts.ContractNumber, Contracts.ContractDate, Supplied.Product, Supplied.Amount, Suppliers.SupplierName (рисунок 3.11);

5) для поля «Supplied.Amount» в рядку «Criteria» ввести Expression: (SELECT MAX (Amount) FROM Supplied) (рисунок 3.11);

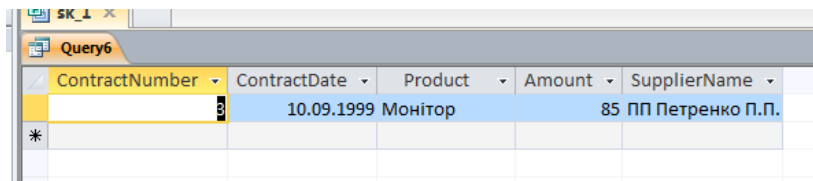
6) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
SELECT    Contracts.ContractNumber,    Contracts.ContractDate,
Supplied.Product, Supplied.Amount, Suppliers.SupplierName
FROM (Suppliers INNER JOIN Contracts ON Suppliers.SupplierID
= Contracts.SupplierID) INNER JOIN Supplied ON
Contracts.ContractNumber = Supplied.ContractNumber
WHERE (((Supplied.Amount)=(SELECT MAX (Amount) FROM
Supplied)));
```

7) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

8) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок

3.12).



ContractNumber	ContractDate	Product	Amount	SupplierName
*	10.09.1999	Монітор	85	ПП Петренко П.П.

Рисунок 3.12 – Режим «DataSheet view»

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

9) закрити вікно запиту і зберегти запит з ім'ям «Запит6».

7. Вивести на екран список постачальників (найменування та код), з якими не було укладено жодного договору.

Рішення:

Варіант 1

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;

3) у вікні «Show table» вибрати і додати за допомогою кнопки «Add» таблицю «Suppliers». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунок 3.13, а);

4) сформуванати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Suppliers.SupplierName, Suppliers.SupplierId (рисунок 3.13, а);

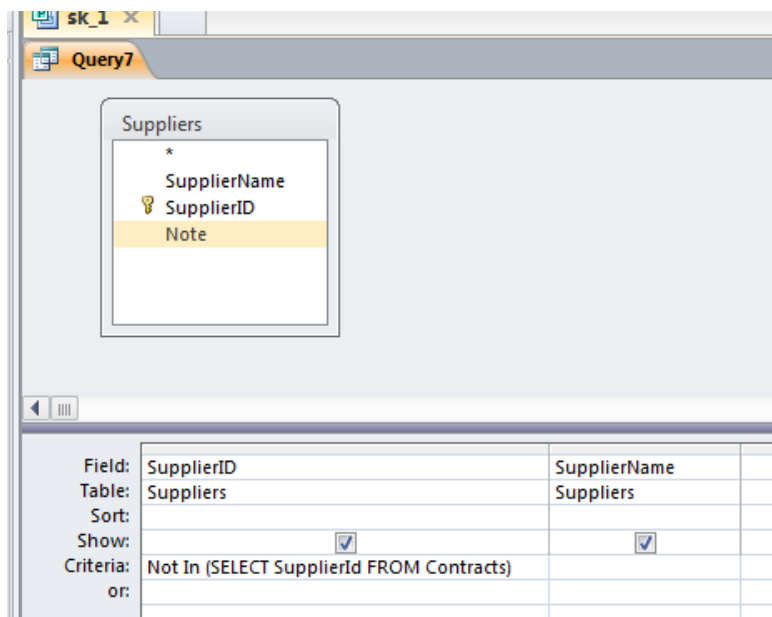


Рисунок 3.13, а – Режим «Query designer»

5) для поля «Suppliers.SupplierId» в рядку «Criteria» ввести вираз: Not In (SELECT SupplierId FROM Contracts) (рисунок 3.13, а);

6) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

SELECT Suppliers.SupplierName, Suppliers.SupplierId FROM Suppliers WHERE (((Suppliers.SupplierId) Not In (SELECT SupplierId FROM Contracts)));

7) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

8) переглянути результат запити, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок 3.14)

SupplierID	SupplierName
4	ЗАТ «Транссервіс»
5	ПП Сидорчук М.С.
*	

Рисунок 1.14 – Режим «DataSheet view»

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

9) закрити вікно запиту і зберегти запит з Ім'ям «Запит7».

Варіант 2

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;

3) у вікні «Show table» вибрати і Add за допомогою кнопки «Add» таблицю «Suppliers». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунок 3.15, б);

4) сформувати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля) Suppliers.SupplierName, Suppliers.SupplierId (Рисунок 3.15, б);

5) для поля «Suppliers.SupplierId» в рядку «Criteria» ввести вираз: <> Any (SELECT SupplierId FROM Contracts) (рисунок 3.15, б);

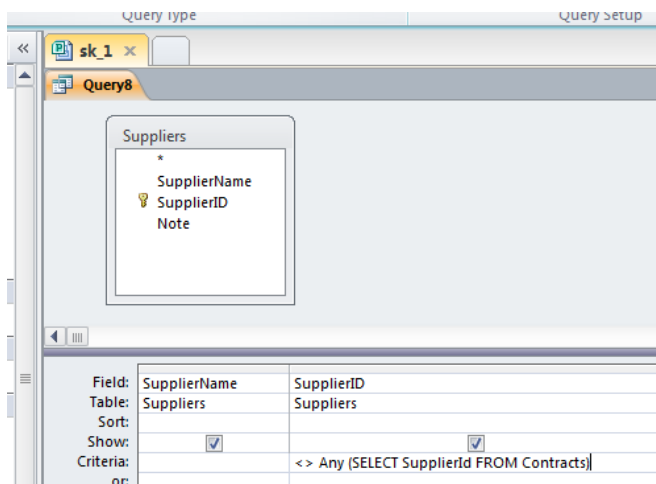


Рисунок 3.15, б – Режим «Query designer»

6) переглянути текст команди SELECT - SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
SELECT Suppliers.SupplierName, Suppliers.SupplierID FROM
Suppliers
WHERE (((Suppliers.SupplierID) <> Any (SELECT SupplierID
FROM Contracts)));
```

7) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

8) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок 3.16).

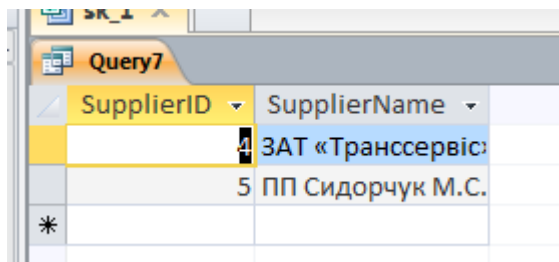


Рисунок 3.16 – Режим «DataSheet view».

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

9) закрити вікно запиту і зберегти запит з Ім'ям «Запит7_1».

Увага! Обидва варіанти, реалізовані при побудові запиту 7, повинні дати однаковий результат. Однак в деяких випадках (як правило, це пов'язано з використанням різних версій СУБД Access), другий варіант може дати неправильний результат (наприклад, вивести список всіх постачальників).

В цьому випадку текст команди SELECT-SQL необхідно змінити наступним чином:

```
SELECT Suppliers.SupplierName, Suppliers.SupplierId
FROM Suppliers
WHERE (not ((Suppliers.SupplierId) = Any (SELECT SupplierId
FROM Contracts))));
```

Далі слід перевірити роботу зміненого запиту і переконатися в тому, результати першого і другого варіанту запиту однакові.

8. Вивести на екран список найменувань поставлених продуктів із зазначенням середньої ціни поставки за одиницю (незалежно від постачальника).

Рішення:

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;

3) у вікні «Show table» вибрати і Add за допомогою кнопки «Add» таблицю

«Supplied». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунок 3.17);

4) сформуванати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Supplied.Product, Supplied.AmountPricePerItem (рисунок 3.17);

5) натиснути правою кнопкою миші в довільному місці таблиці зі списком солей запиту і увімкнути режим «Групові операції», для поля «Product» встановити режим «Group by», для поля «AmountPricePerItem» – «Avg» (рисунок 3.17);

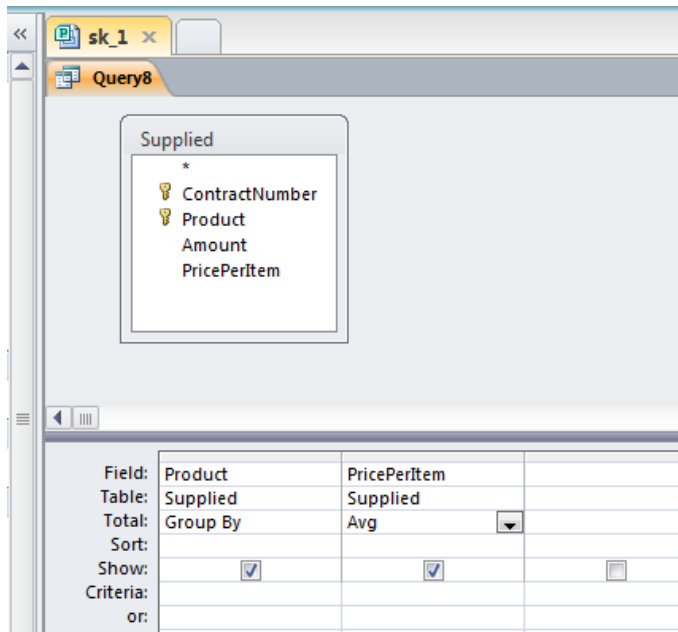


Рисунок 3.17 – Режим «Query designer»

6) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

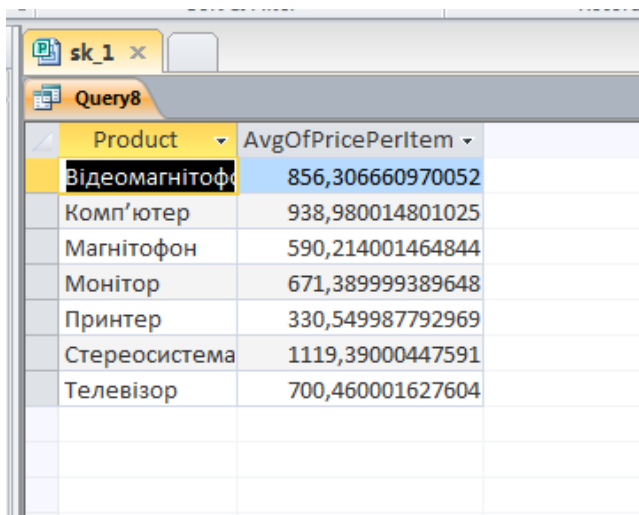
```
SELECT Supplied.Product, Avg(Supplied.PricePerItem) AS
AvgOfPricePerItem
```

```
FROM Supplied
```

```
GROUP BY Supplied.Product;
```

7) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

8) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view».



Product	AvgOfPricePerItem
Відеомагнітоф	856,306660970052
Комп'ютер	938,980014801025
Магнітофон	590,214001464844
Монітор	671,389999389648
Принтер	330,549987792969
Стереосистема	1119,39000447591
Телевізор	700,460001627604

Рисунок 3.18 – Режим «DataSheet view»

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

9) закрити вікно запиту і зберегти запит з ім'ям «Запит8».

9. Вивести на екран список продуктів (найменування, кількість і ціна, постачальник), для яких ціна за одиницю більше середньої.

Рішення:

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;

3) у вікні «Show table» вибрати і Add по черзі за допомогою кнопки «Add» таблиці «Contracts», «Supplied», «Suppliers». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунок 3.9);

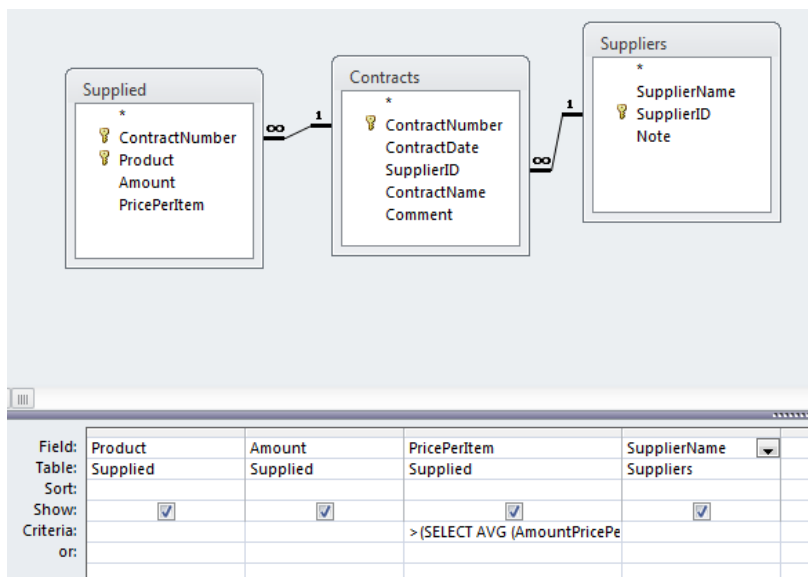


Рисунок 3.19 – Режим «Query designer»

4) сформуванати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Supplied.Product, Supplied.Amount, Supplied.AmountPricePerItem, Suppliers.SupplierName (рисунок 3.19);

5) для поля «Supplied.AmountPricePerItem» в рядку «Criteria» ввести вираз:

> (SELECT AVG (AmountPricePerItem) FROM Supplied) (рисунок 3.19);

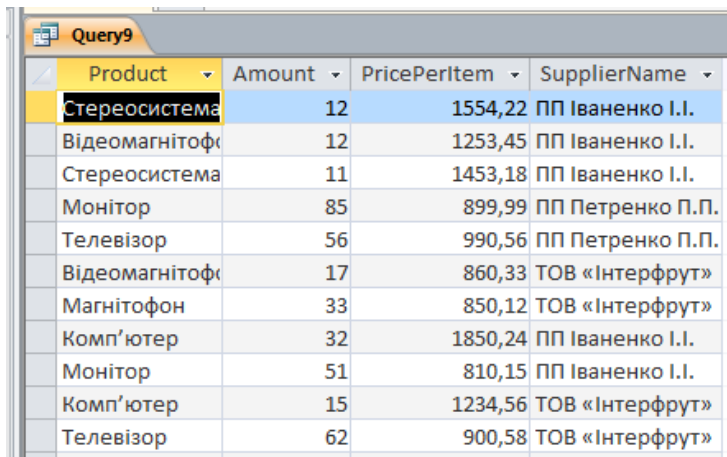
6) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
SELECT      Supplied.Product,      Supplied.Amount,
Supplied.AmountPricePerItem, Suppliers.SupplierName FROM Suppliers
INNER JOIN (Contracts INNER JOIN Supplied ON
Contracts.ContractNumber = Supplied.ContractNumber) ON
Suppliers.SupplierId = Contracts.SupplierId WHERE
(((Supplied.AmountPricePerItem)> (SELECT AVG (AmountPricePerItem)
FROM Supplied)))
```

7) натиснути правою кнопкою миші по заголовку вікна і в меню

вибрати режим «Query designer»;

8) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок 3.20).



Product	Amount	PricePerItem	SupplierName
Стереосистема	12	1554,22	ПП Іваненко І.І.
Відеомагнітофон	12	1253,45	ПП Іваненко І.І.
Стереосистема	11	1453,18	ПП Іваненко І.І.
Монітор	85	899,99	ПП Петренко П.П.
Телевізор	56	990,56	ПП Петренко П.П.
Відеомагнітофон	17	860,33	ТОВ «Інтерфрут»
Магнітофон	33	850,12	ТОВ «Інтерфрут»
Комп'ютер	32	1850,24	ПП Іваненко І.І.
Монітор	51	810,15	ПП Іваненко І.І.
Комп'ютер	15	1234,56	ТОВ «Інтерфрут»
Телевізор	62	900,58	ТОВ «Інтерфрут»

Рисунок 3.20 – Режим «DataSheet view»

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

9) закрити вікно запиту і зберегти запит з ім'ям «Запит9».

10. Вивести на екран відомості про п'ять найдорожчих продуктів (найменування, ціна за одиницю, постачальник).

Рішення:

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «ОК»;

3) у вікні «Show table» вибрати і Add по черзі за допомогою кнопки «Add» таблиці «Contracts», «Supplied», «Suppliers». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунок 3.21);

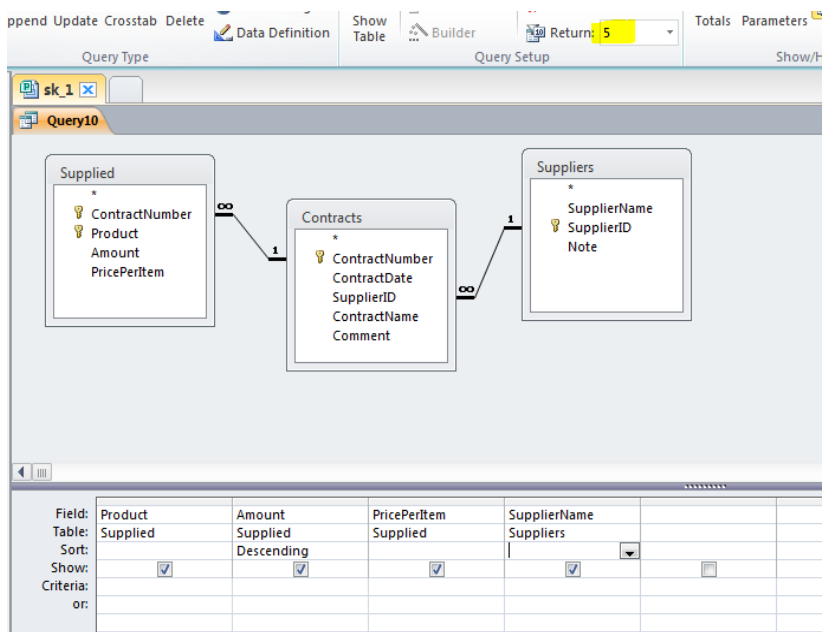


Рисунок 3.21 – Режим «Query designer»

4) сформувати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Supplied.Product, Supplied.AmountPricePerItem, Suppliers.SupplierName (рисунок 3.20);

5) для поля «Supplied.PricePerItem» в рядку «Sort» встановити режим «By desc» (рисунок 3.20);

6) за допомогою списку «return» встановити значення «5», що визначає кількість відображаючих записів (рисунок 3.21);

7) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

SELECT TOP 5 Supplied.Product, Supplied.AmountPricePerItem, Suppliers.SupplierName FROM (Suppliers INNER JOIN Contracts ON Suppliers.SupplierId = Contracts.SupplierId) INNER JOIN Supplied ON Contracts.ContractNumber = Supplied.ContractNumber ORDER BY Supplied.AmountPricePerItem DESC;

8) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

9) переглянути результат запиту, для чого натиснути правою

кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок 3.21).

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

Product	Amount	PricePerItem	SupplierName
Монітор	85	899,99	ПП Петренко П.П.
Телевізор	62	900,58	ТОВ «Інтерфрут»
Телевізор	56	990,56	ПП Петренко П.П.
Телевізор	52	545,32	ПП Петренко П.П.
Монітор	51	810,15	ПП Іваненко І.І.
*			

Рисунок 3.21 – Режим «Query designer»

10) закрити вікно запиту і зберегти запит з ім'ям «Запит10».

11. Для кожного дня 9 місяці 1999 року визначити, на яку суму були поставлені продукти кожним постачальником (в результат запиту включати тільки ті дні, в які були поставки продуктів).

Рішення:

- 1) натиснути кнопку «Create»;
- 2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;
- 3) у вікні «Show table» вибрати і Add по черзі за допомогою кнопки «Add» таблиці «Contracts», «Supplied», «Suppliers», вікно «Show table» закрити.

Внаслідок на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунок 3.22);

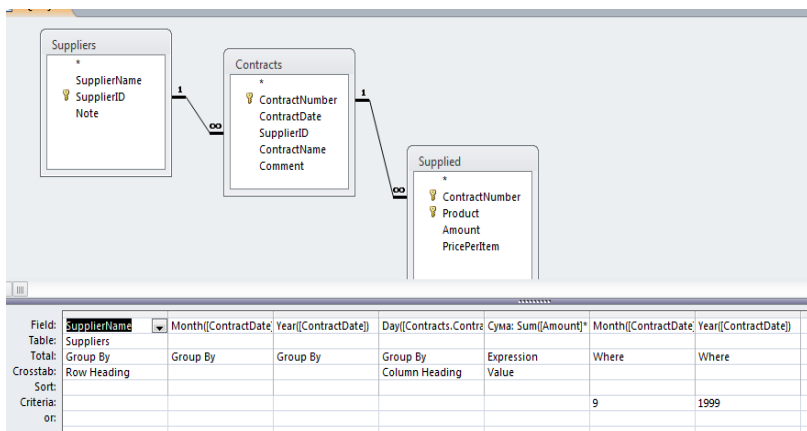


Рисунок 3.22 – Режим «Query designer»

4) натиснути правою кнопкою миші і меню вибрати пункт «Query type». У підменю вибрати тип «CrossTabQuery»;

5) сформуванати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Suppliers.SupplierName, День: DAY ([Contracts.ContractDate]), Сума: Sum ([Amount] * [AmountPricePerItem]) (рисунок 3.22). Поле «Сума» є обчислюваним, а поле «День» – виразом, і тому для них імена таблиць не вказуються;

6) сформуванати список полів, що забезпечують фільтрацію даних в результаті запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Місяць: Month ([ContractDate]), Рік: Year ([ContractDate]);

7) для поля «Сума» в рядку «Total» встановити операцію «Sum» (малюнок 1.22);

8) в рядку «CrossTab» для поля SupplierName встановити режим «Row Heading», для поля «День» – «Column Heading», для поля «Сума» – «Value»;

9) для полів «Місяць» і «Рік» в рядку «Criteria» встановити значення 9 і 1999 відповідно;

10) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
TRANSFORM Sum ([Amount] * [AmountPricePerItem]) AS Сума
SELECT Suppliers.SupplierName
FROM Suppliers INNER JOIN (Contracts INNER JOIN Supplied
```

ON Contracts.ContractNumber = Supplied.ContractNumber) ON
 Suppliers.SupplierId = Contracts.SupplierId
 WHERE (((Month ([ContractDate])) = 9) AND ((Year
 ([ContractDate])) = 1999))
 GROUP BY Suppliers.SupplierName, Month ([ContractDate]), Year
 ([ContractDate]) PIVOT Day ([ContractDate]);

11) натиснути правою кнопкою миші по заголовку вікна і в
 меню вибрати режим «Query designer»;

12) переглянути результат запиту, для чого натиснути правою
 кнопкою миші і в меню вибрати пункт «DataSheet view».

SupplierName	1	10	23	24
ПП Іваненко І.І.	63592,2893676758	41562,0612487793		
ПП Петренко П.П.		110839,789550781	85604,879119873	
ТОВ «Інтерфрут»				76712,2691040039

Рисунок 3.23 – Режим «DataSheet view»

Кладнути правою кнопкою миші по заголовку вікна і в меню
 вибрати режим «Query designer»;

13) закрити вікно запиту і зберегти запит з Ім'ям «Запит11».

12. Сформувані список договорів (тільки номери), загальну
 кількість поставлених продуктів і загальну суму за кожним договором.
 Для постачальників – фізичних осіб вивести LastName, FirstName, по
 SecondName (повністю), для постачальників – юридичних осіб – номер
 свідоцтва платника ПДВ.

Рішення:

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «Query designer» і натиснути
 кнопку «OK»;

3) у вікні «Show table» вибрати по черзі і Add за допомогою
 кнопки «Add» таблиці «Contracts», «Supplied»,
 «IndividualEntrepreneurs», «LegalEntities». Вікно «Show table» закрити.
 В результаті на екрані з'явиться вікно для побудови запиту, де буде
 видно вибрані таблиці і зв'язки між ними (рисунок 3.24);

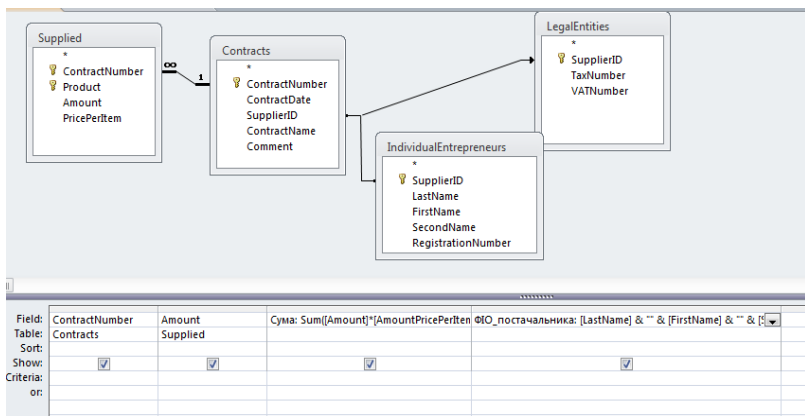


Рисунок 3.24 – Режим «Query designer»

4) сформуванати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Contracts.ContractNumber, Supplied.Amount, сума: Sum ([Amount] * [AmountPricePerItem]), ФІО_постачальника: [LastName] & " " & [FirstName] & " " & [SecondName], RegistrationNumber (рисунок 3.24). Поле «Сума» є обчислюваним, а поле «ФІО_постачальника» – виразом, і тому для них імена таблиць не вказуються;

5) для поля «ContractNumber» встановити в рядку «Sort» режим «Asc»;

6) натиснути правою кнопкою миші в довільному місці таблиці зі списком полів запиту і включити режим «Totals». Для полів «ContractNumber» і «ContractDate» встановити режим «Group by», для поля «Amount» – «Sum», для поля Сума – «Expression» (малюнок 1.24);

7) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view». Запит не повинен дати результату! Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

8) клацнути мишею по зв'язку між таблицями «Contracts» і «IndividualEntrepreneurs». У вікні

«Join properties» встановити режим 2 – Об'єднання BCIX записів з «Contracts» і тільки тих записів з «IndividualEntrepreneurs», в яких пов'язані поля збігаються;

9) клацнути мишею по зв'язку між таблицями «Contracts» і

«LegalEntities». У вікні «Join properties» встановити режим 2 – Об'єднання BCIX записів з «Contracts» і тільки тих записів з «LegalEntities», в яких пов'язані поля збігаються;

10) Переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
SELECT Contracts.ContractNumber, Sum (Supplied.Amount) AS
[Sum-Amount], Sum ([Amount] * [AmountPricePerItem]) AS Сума,
[LastName] & "" & [FirstName] & "" & [SecondName] AS
ФІО_постачальника , LegalEntities.RegistrationNumber
```

```
FROM ((Contracts LEFT JOIN IndividualEntrepreneurs ON
Contracts.SupplierId = IndividualEntrepreneurs.SupplierId) LEFT JOIN
LegalEntities ON Contracts.SupplierId = LegalEntities.SupplierId) INNER
JOIN Supplied ON Contracts.ContractNumber = Supplied.ContractNumber
GROUP BY Contracts.ContractNumber, [LastName] & "" &
[FirstName] & "" & [SecondName], LegalEntities.RegistrationNumber
ORDER BY Contracts.ContractNumber;
```

11) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

12) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view». Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

13) закрити вікно запиту і зберегти запит з ім'ям «Запит12».

13. Визначити, яку кількість кожного продукту було потавлено кожним постачальником.

Рішення:

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;

3) у вікні «Show table» вибрати і Add по черзі за допомогою кнопки «Add» таблиці «Contracts», «Supplied», «Suppliers». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрані таблиці і зв'язки між ними (рисунк 3.25);

4) натиснути правою кнопкою миші і меню вибрати пункт

«Query type». У підменю вибрати тип «CrossTab»;

5) сформувати список полів, які потраплять в результат запиту. Це такі поля (Ім'яТаблиці.Ім'яПоля): Suppliers.SupplierName, Supplied.Product, Supplied.Amount (рисунок 3.25);

6) для поля «Сума» в рядку «Group by» встановити операцію «Sum» (рисунок 3.25);

7) в рядку «CrossTab» для поля «SupplierName» встановити режим «Row header», для поля «Product» – «Заголовки стовпців», для поля «Amount» – «Value»;

8) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
TRANSFORM Sum (Supplied.Amount) AS [Sum-Amount]
SELECT Suppliers.SupplierName
FROM Suppliers INNER JOIN (Contracts INNER JOIN Supplied
ON Contracts.ContractNumber = Supplied.ContractNumber) ON
Suppliers.SupplierId = Contracts.SupplierId
GROUP BY Suppliers.SupplierName PIVOT Supplied.Product;
```

9) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

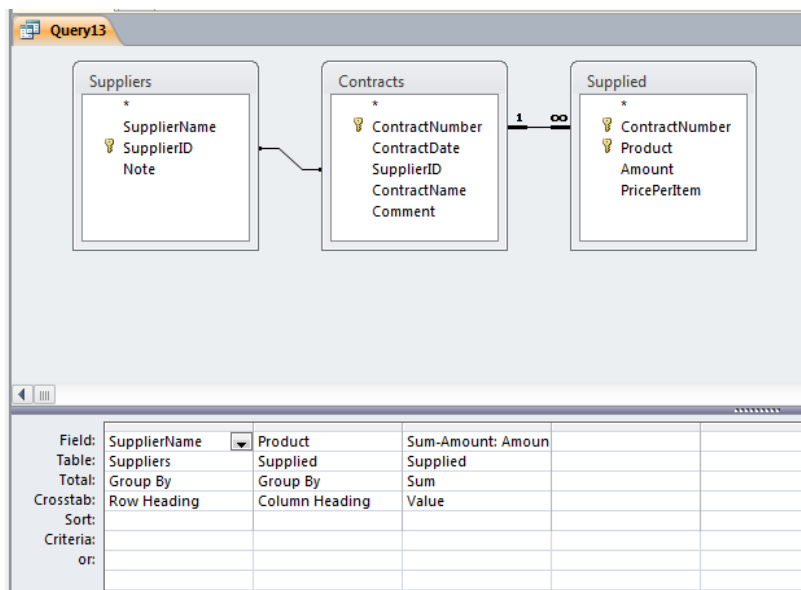


Рисунок 3.25 – Режим Query Designer

10) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок 3.26).

SupplierName	Відеомагніт	Комп'ютер	Магнітофон	Монітор	Принтер	Стереосист	Телевізор
ПІ Іваненко І.І.	20	99	30	51		23	44
ПІ Петренко П.П.			33	85	41	27	108
ТОВ «Інтерфрут»	17	15	33	66			76

Рисунок 3.26 – Режим «DataSheet view»

Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

11) закрити вікно запиту і зберегти запит з Ім'ям «Запит13».

14. Вивести на екран список договорів (номер, дата) і загальну суму за кожним договором (розмір партії помножити на ціну за штуку і підсумувати за договором). Список повинен бути відсортований в порядку зростання загальних сум по кожному договору. Крім того, на список має бути накладено умова фільтрації, що складається у

виключенні з результату запиту записів, для яких ContractNumber більше зазначеного значення.

Коментар до умови запиту.

Умова запиту аналогічно умові запиту 5. Відмінність полягає в тому, що значення номера договору, на підставі якого виконується фільтрація записів, не вказується в запиті явно, а має бути введені при запуску запиту. Таке значення називається параметром запиту.

Рішення:

1) повторити дії, описані при створенні запиту 5, крім установки умовифільтрації;

2) після створення запиту натиснути правою кнопкою миші і в меню вибрати пункт «Parameters»;

3) в таблиці вікна «Query parameters», в колонці «Parameter» ввести назву параметра – «Nomer» (без лапок), в колонці «Data type» встановити тип «Ціле». Закрити вікно «Parameters», натиснувши кнопку «ОК»;

4) в рядку «Criteria» ввести вираз $n > [nomer]$ (рисунок 3.26);

5) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
PARAMETERS nomer Short;
SELECT Contracts.ContractNumber, Contracts.ContractDate, Sum
(Supplied.Amount) AS Sum_Amount, Sum ([Amount] *
[AmountPricePerItem]) AS Загальна сума
FROM Contracts INNER JOIN Supplied ON
Contracts.ContractNumber = Supplied.ContractNumber
GROUP BY Contracts.ContractNumber, Contracts.ContractDate
HAVING (((Contracts.ContractNumber) > [nomer]))
ORDER BY Contracts.ContractNumber;
```

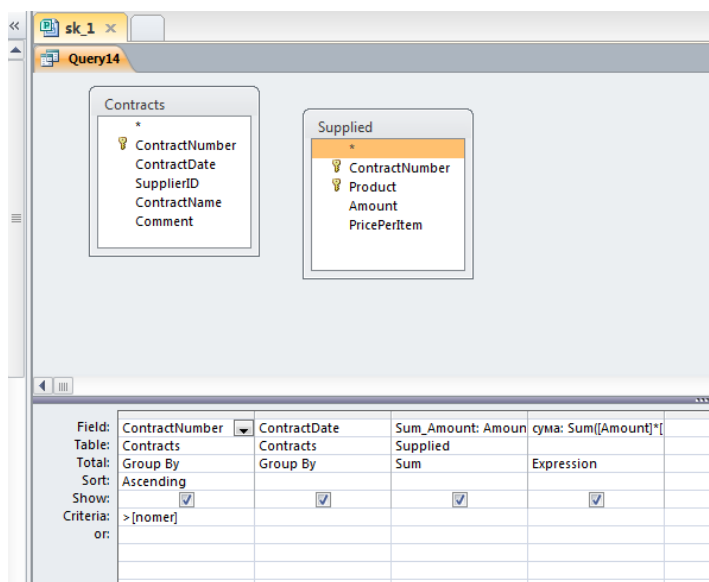


Рисунок 3.26 – Режим «Query designer»

6) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

7) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (Рисунок 3.27, параметри 5, 12)

Query14				
ContractNumber	ContractDate	Sum_Amount	сума	
5	01.10.1999	768	9216	
7	02.10.1999	768	9216	

Рисунок 3.27 – Режим «DataSheet view»

Ввести значення параметра. Переглянути результат кілька разів з різними значеннями параметра. Клацнути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

8) закрити вікно запиту і зберегти запит «Запит14».

15. Сформуванати список продуктів, поставлених

постачальниками 1 (ПП Петров П.П.) та 2 (ТОВ «Інтерфрут»).

Коментар до умови запиту.

Даний запит використовується як ілюстрація застосування операції UNION – об'єднання таблиць, одержуваних в результаті виконання декількох запитів. Необхідно відзначити, що для вирішення даного завдання використовувати операцію UNION не обов'язково.

Рішення:

- 1) натиснути кнопку «Create»;
- 2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;
- 3) вікно «Show table» закрити, не додаючи жодної таблиці;
- 4) натиснути правою кнопкою миші і меню вибрати пункт «Запит SQL». У підменю вибрати пункт «Union»;
- 5) ввести текст запиту:

```
SELECT Product, Amount, AmountPricePerItem, Amount *  
AmountPricePerItem AS Сума, SupplierName FROM Contracts, Suppliers,  
Supplied WHERE Contracts.ContractNumber = Supplied.ContractNumber  
AND Contracts.SupplierId = Suppliers.SupplierId AND  
Suppliers.SupplierId = 1
```

UNION

```
SELECT Product, Amount, AmountPricePerItem, Amount *  
AmountPricePerItem AS Сума, SupplierName FROM Contracts, Suppliers,  
Supplied WHERE Contracts.ContractNumber = Supplied.ContractNumber  
AND Contracts.SupplierId = Suppliers.SupplierId AND  
Suppliers.SupplierId = 2;
```

Зовнішній вигляд запиту наведено на рисунку 1.28;

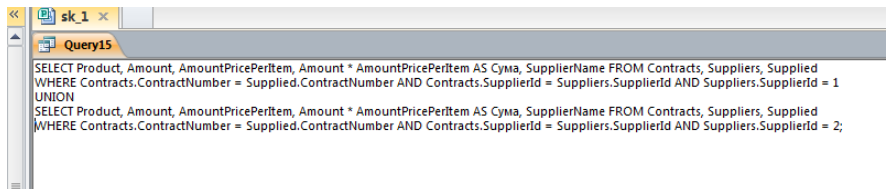
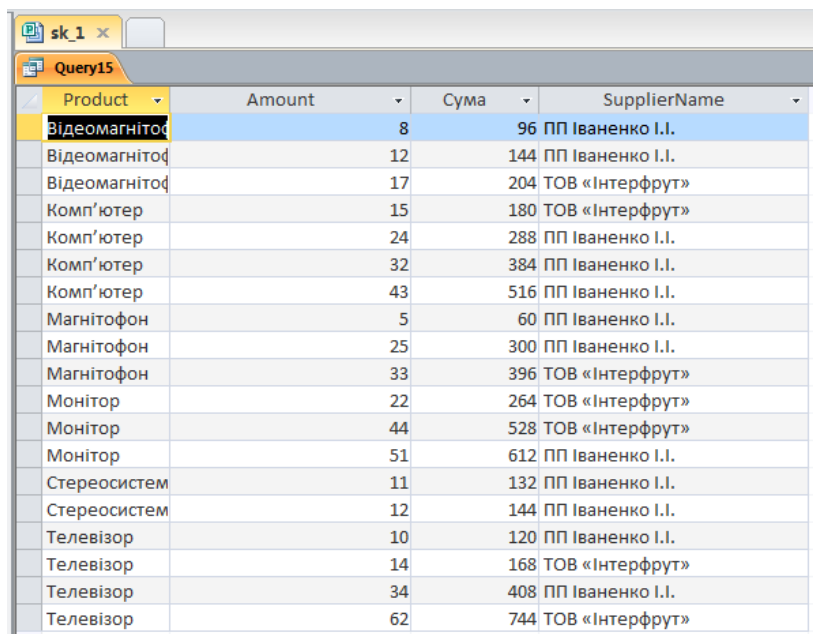


Рисунок 3.28 –Зовнішній вигляд запиту

- б) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок

3.29);



Product	Amount	Сума	SupplierName
Відеомагнітоф	8	96	ПП Іваненко І.І.
Відеомагнітоф	12	144	ПП Іваненко І.І.
Відеомагнітоф	17	204	ТОВ «Інтерфрут»
Комп'ютер	15	180	ТОВ «Інтерфрут»
Комп'ютер	24	288	ПП Іваненко І.І.
Комп'ютер	32	384	ПП Іваненко І.І.
Комп'ютер	43	516	ПП Іваненко І.І.
Магнітофон	5	60	ПП Іваненко І.І.
Магнітофон	25	300	ПП Іваненко І.І.
Магнітофон	33	396	ТОВ «Інтерфрут»
Монітор	22	264	ТОВ «Інтерфрут»
Монітор	44	528	ТОВ «Інтерфрут»
Монітор	51	612	ПП Іваненко І.І.
Стереосистем	11	132	ПП Іваненко І.І.
Стереосистем	12	144	ПП Іваненко І.І.
Телевізор	10	120	ПП Іваненко І.І.
Телевізор	14	168	ТОВ «Інтерфрут»
Телевізор	34	408	ПП Іваненко І.І.
Телевізор	62	744	ТОВ «Інтерфрут»

Рисунок 3.29 – Режим «DataSheet view»

7) закрити вікно запиту і зберегти запит з ім'ям «Запит15».

16. Сформувати список продуктів, які поставлялися більше 1 разу. Рішення:

Варіант 1

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;

3) у вікні «Show table» вибрати і Add за допомогою кнопки «Add» таблицю «Supplied». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрану таблицю (рисунок 3.30).

4) сформувати список полів, які потраплять в результат запиту. Для цього двічі вибрати поле «Product» (малюнок 1.30). Для першого з цих полів ввести умову відбору (рисунок 3.30):

In (SELECT Product FROM Supplied GROUP BY Product HAVING COUNT (Product)> 1)

Включити перемикач «Totals» і для другого з полів Product встановити операцію «Count» (рисунок 3.30);

5) переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
SELECT Supplied.Product, Count (Supplied.Product) AS [Count-Product] FROM Supplied
GROUP BY Supplied.Product
HAVING (((Supplied.Product) In (SELECT Product FROM Supplied GROUP BY Product HAVING COUNT (Product)> 1)));
```

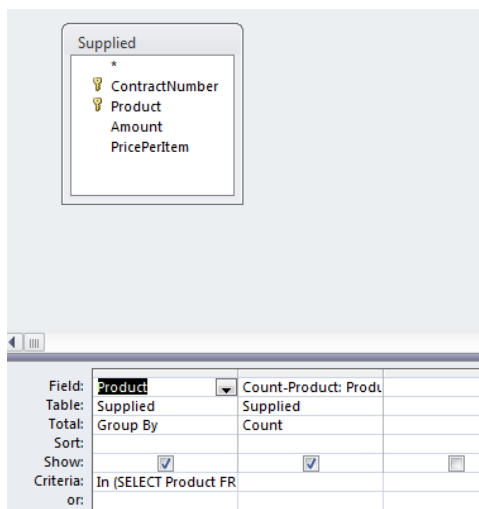
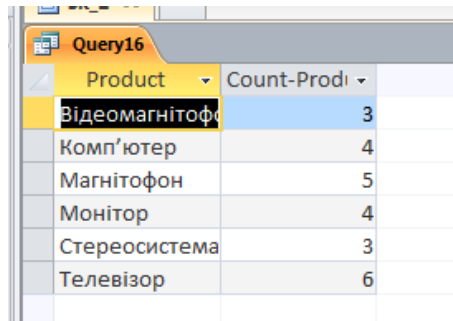


Рисунок 3.30 – Режим «Query designer»

6) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view»;



Product	Count-Product
Відеомагнітоф	3
Комп'ютер	4
Магнітофон	5
Монітор	4
Стереосистема	3
Телевізор	6

Рисунок 3.31 – Режим «DataSheet view»

7) закрити вікно запиту і зберегти запит з ім'ям «Запит16».

Варіант 2

1) натиснути кнопку «Create»;

2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;

3) у вікні «Show table» вибрати і Add за допомогою кнопки «Add» таблицю «Supplied». Вікно «Show table» закрити. В результаті на екрані з'явиться вікно для побудови запиту, де буде видно вибрана таблиця (рисунок 3.32);

4) сформувавати список полів, які потраплять в результат запиту. Для цього двічі вибрати поле «Product» (малюнок 1.32). Включити перемикач «Totals» і для другого з полів «Product» встановити операцію «Count». Для нього ввести умову відбору: > 1 (рисунок 3.32):

5) Переглянути текст команди SELECT-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
SELECT Supplied.Product, Count (Supplied.Product) AS [Count-Product] FROM Supplied
GROUP BY Supplied.Product HAVING COUNT (Product)> 1;
```

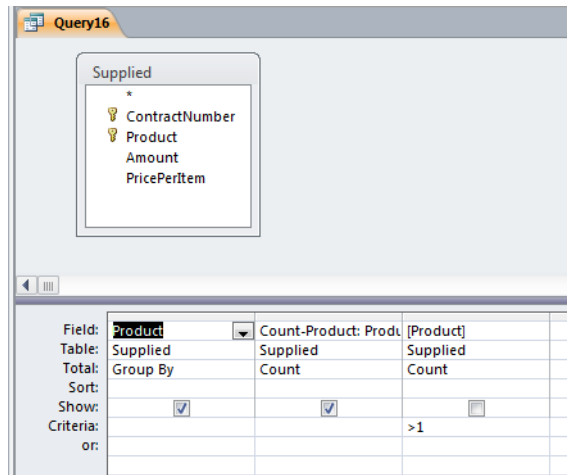


Рисунок 3.32 – Режим «Query designer»

б) переглянути результат запиту, для чого натиснути правою кнопкою миші і в меню вибрати пункт «DataSheet view» (рисунок 3.33);

Product	Count-Product
Відеоманітоф	3
Комп'ютер	4
Манітофон	5
Монітор	4
Стереосистема	3
Телевізор	6

Рисунок 3.33 – Режим «DataSheet view»

7) закрити вікно запиту і зберегти запит з ім'ям «Запит16_2».

II. Маніпулювання даними за допомогою команд UPDATE-SQL, DELETE-SQL.

Увага! Наведені нижче результати виконання UPDATE-SQL запити відповідають стану бази даних, яка була створена у першій

лабораторній роботі.

17. Збільшити на 10 одиниць кількість кожного продукту, поставленого постачальником 1 (ПП Петров П.П.).

Рішення:

- 1) натиснути кнопку «Create»;
- 2) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;
- 3) у вікні «Show table» вибрати і Add за допомогою кнопки «Add» таблицю «Supplied». Вікно «Show table» закрити;
- 4) натиснути правою кнопкою миші і меню вибрати пункт «Query type». У підменю вибрати тип «Update»;
- 5) сформувати список полів, які використовуються при оновленні даних. Це такі поля (Ім'яТаблиці.Ім'яПоля): Supplied.Amount, Supplied.ContractNumber (рисунок 3.34);
- 6) для поля «Supplied.ContractNumber» в рядку «Criteria» ввести вираз «In (SELECT ContractNumber FROM Contracts WHERE SupplierId = 1)» (без лапок) (рисунок 3.34);
- 7) для поля «Supplied.Amount » в рядку «Update» ввести вираз «[Supplied].[Amount] +10» (без лапок) (рисунок 3.34);
- 8) переглянути текст команди UPDATE-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:

```
UPDATE Supplied SET Supplied.Amount = Supplied.Amount + 10
WHERE (((Supplied.ContractNumber) In (SELECT
ContractNumber FROM Contracts WHERE SupplierId = 1)))
```
- 9) натиснути правою кнопкою миші по заголовку вікна і в меню вибрати режим «Query designer»;

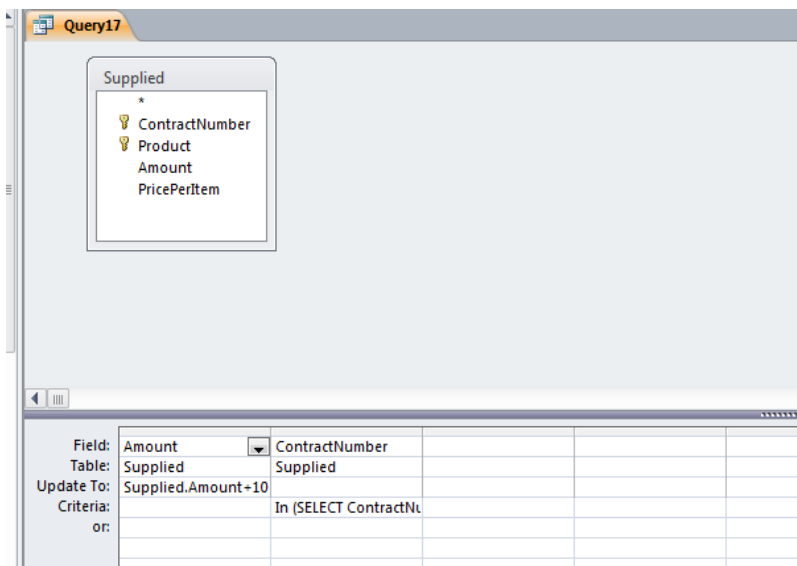


Рисунок 3.34 – Режим «Query designer»;

- 10) в панелі інструментів натиснути кнопку «execute» і підтвердити оновлення даних;
- 11) відкрити таблицю «Supplied» і переконатися в правильності поновлення даних;
- 12) закрити вікно запиту і зберегти запит з ім'ям «Запит17».

Увага! Наведені нижче результати виконання DELETE-SQL запиту відповідають стану бази даних, яка була створена у першій лабораторній роботі.

18. Видалити всі «порожні» контракти (тобто ті контракти, для яких відсутні записи про поставлені продукти).

Рішення:

- 1) відкрити вкладку «Таблиці» вікна бази даних;
- 2) відкрити таблицю «Contracts»;
- 3) Додати новий запис. Ввести до цього запису наступні дані:

ContractNumber	-	8
ContractDate	-	27.07.2002
SupplierId	-	3

- 4) закрити таблицю «Contracts»;
- 5) відкрити вкладку «Queries» вікна бази даних;
- 6) натиснути кнопку «Create»;
- 7) у вікні «Create» вибрати режим «Query designer» і натиснути кнопку «OK»;
- 8) у вікні «Show table» вибрати і Add за допомогою кнопки «Add» таблицю «Contracts». Вікно «Show table» закрити;
- 9) натиснути правою кнопкою миші і меню вибрати пункт «Query type». У підменю вибрати тип «Видалення»;
- 10) визначити поле, яке використовується при видаленні даних в якості ключового. Це поле Contracts.ContractNumber (рисунок 3.35);
- 11) для поля «Contracts.ContractNumber» в рядку «Criteria» ввести вираз «Not In (SELECT ContractNumber FROM Supplied)» (без лапок) (рисунок 3.35);
- 12) переглянути текст команди DELETE-SQL. Для цього натиснути правою кнопкою миші і в меню вибрати пункт «SQL view». У вікні буде приведений наступний текст:
DELETE Contracts.ContractNumber FROM Contracts
WHERE (((Contracts.ContractNumber) Not In (SELECT ContractNumber FROM Supplied)));

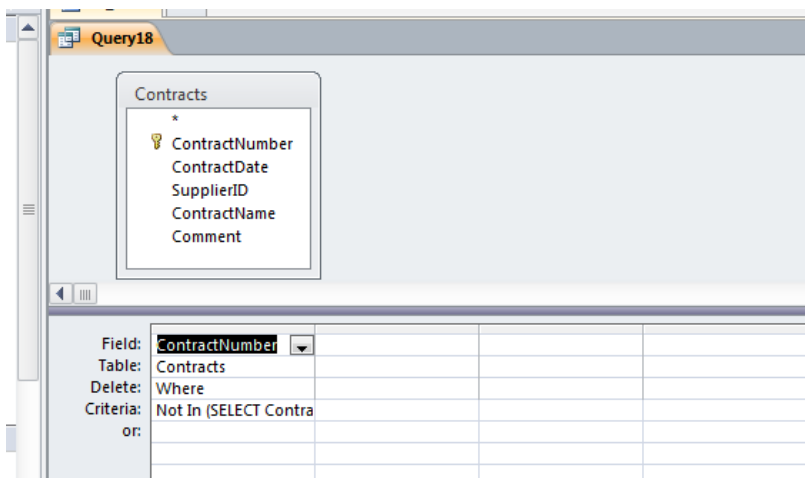


Рисунок 3.35 – Режим «Query designer»

- 13) натиснути правою кнопкою миші по заголовку вікна і в

меню вибрати режим «Query designer»;

14) в панелі інструментів натиснути кнопку «Запуск» і підтвердити видалення даних;

15) відкрити таблицю «Contracts» і переконатися в правильності видалення даних;

16) закрити вікно запиту і зберегти запит з Ім'ям «Запит18».

III. Закінчити роботу. Зберегти файл SK.mdb.

4 ВИМОГИ ДО ЗВІТУ

Відповіднім чином оформлений та роздрукований звіт з лабораторної роботи є документом, що підтверджує виконання студентом лабораторної роботи.

У звіті з лабораторної роботи:

- 1) коротко описати основні етапи виконання завдання;
- 2) навести скріншоти створення таблиць бази даних у режимі таблиці та режимі конструктора;
- 3) відобразити структуру створеної бази даних та відношень між таблицями;
- 4) описати результати дослідження, особливості роботи відношень посилальної цілісності, встановлені між таблицями.

Звіт з лабораторної роботи роздруковується на аркуші формату А4, він повинен мати відповідний титульний аркуш. Роздрукований звіт здається студентом викладачеві у файлі.

Звіт має бути оформлений за такими вимогами:

- параметри сторінки: лівий відступ – 3 см; правий – 1,5 см; верхній та нижній відступи по 2 см;
- шрифт Times New Roman, 14;
- налаштування абзацу: вирівнювання – за шириною, відступи зліва та справа – 0 см., відступ першого рядка – 1,25 см, інтервал перед та після абзацу – 0 пт, міжрядковий інтервал – одинарний; на вкладці «Положення на сторінці» відключити функцію «Заборона висячих рядків».

Усі скріншоти, розміщені у звіті, оскільки є рисунками, повинні мати підписи та відповідну нумерацію.

5 ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Команда SELECT-SQL. Загальна характеристика, призначення та використання.
2. Команда SELECT-SQL. Реляційні операції, які реалізуються за допомогою команди SELECT-SQL.
3. Команда SELECT-SQL. Стовпці виведення результату запиту. Використання виразів в шпальтах виведення.
4. Команда SELECT-SQL. Вираз FROM. Призначення і використання.
5. Команда SELECT-SQL. Вираз WHERE. Призначення і використання.
6. Команда SELECT-SQL. Вираз WHERE. FILTER-умови і їх використання.
7. Команда SELECT-SQL. Вираз WHERE. Формування запиту на базі декількох таблиць. JOIN-умови і їх використання.
8. Команда SELECT-SQL. З'єднання таблиць в багатотабличних запитах. Види з'єднань – INNER JOIN, LEFT JOIN, RIGHT JOIN і їх особливості.
9. Команда SELECT-SQL. Перехресні запити і їх особливості.
10. Команда SELECT-SQL. Аргумент DISTINCT. Призначення і використання.
11. Команда SELECT-SQL. Аргумент TOP. Призначення і використання.
12. Команда SELECT-SQL. Булеві оператори AND, OR, NOT та їх використання
13. Команда SELECT-SQL. Спеціальні оператори IN, BETWEEN. Призначення і використання.
14. Команда SELECT-SQL. Агрегує функції COUNT, SUM, AVG, MAX, MIN. Призначення і використання.
15. Команда SELECT-SQL. Пропозиція GROUP BY. Призначення і використання.
16. Команда SELECT-SQL. Пропозиція ORDER BY. Призначення і використання.
17. Команда SELECT-SQL. Пропозиція HAVING. Призначення і використання.
18. Команда SELECT-SQL. Підзапити. Призначення і

використання.

19. Команда SELECT-SQL. Підзапити. Види підзапитів і їх особливості.

20. Команда SELECT-SQL. Параметри і їх використання при побудові запитів.

21. Команда INSERT-SQL. Загальна характеристика, призначення та використання.

22. Команда DELETE-SQL. Загальна характеристика, призначення та використання.

23. Команда UPDATE-SQL. Загальна характеристика, призначення та використання.

24. Як реалізувати запит 15 без використання операції UNION?

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Access. Збірник рецептів для професіоналів. 2-е вид. / К.Гетц, П.Літвін, Е.Берон. – СПб.: Пітер, 2005. – 782 с.
2. Андон Ф., Резніченко В. Мова запитів SQL. Навчальний курс. – СПб.: Пітер; Київ: Видавнича група BHV, 2006. – 416 с.
3. Бекаревич Ю.Б., Пушкіна Н.В. Microsoft Access за 21 заняття для студента. – СПб.: БХВ-Петербург, 2005. – 544 с.
4. Гарсія-Моліна Г., Ульман Д., Уідом Д. Системи баз даних. Повний курс : Пер. з англ. – М.: Видавничий дім «Вільямс», 2004. – 1088 с.
5. Грабер М. Введення в SQL. – М.: Лорі, 1996. – 375 с.
6. Дейт, К. Дж. Введення в системи баз даних. : Пер. з англ. – 6-е изд. – К.: Діалектика, 1998. – 784 с.
7. Дженнінгс Р. Використання Microsoft Office Access 2003. Спеціальне видання: Пер. з англ. – М.: Видавничий дім «Вільямс», 2005. – 1312 с.
8. Кімел П. Освой самостійно програмування для Microsoft Access 2002 за 24 год .: Пер. з англ. – М.: Видавничий дім «Вільямс», 2003. – 480 с.
9. Клайн К. SQL. Довідник. 2-е видання / Пер. з англ. – М.: КУДИЦ-ОБРАЗ, 2006. – 832 с.
10. Коннолі Т., Бегг К., Страчан А. Бази даних: проектування, реалізація і супровід. Теорія і практика., 2-е вид.: Пер. з англ. – М.: Видавничий дім «Вільямс», 2001. – 1120 с.
11. Кренке Д. Теорія і практика побудови баз даних. 9-е изд. – СПб.: Пітер, 2005. – 859 с.
12. Мак-Федрік П. Форми, звіти і запити в Microsoft Access 2003. : Пер. з англ. – М.: Видавничий дім «Вільямс», 2005. – 416 с.
13. Мірошніченко Г.А. Реляційні бази даних: практичні прийоми оптимальних рішень. – СПб.: БХВ-Петербург, 2005. – 400 с.
14. Моїсеєнко С.І. SQL. Завдання і рішення. – СПб.: Пітер, 2006. – 256 с.
15. Роб П., Коронел К. Системи баз даних: проектування, реалізація і управління. – 5-е изд., Перераб. і доп.: Пер. з англ. – СПб.: БХВ-Петербург, 2004. – 1040 с.
16. Селко Д. Програмування на SQL для професіоналів. –

М.: Видавництво «Лорі», 2004. – 442 с.

17. Ситник Н.В., Краснюк М.Т. Проектування баз и сховищ даних: Навч.-метод. посіб. для самост. вивч. дисц. – К.: КНЕУ, 2005. – 264 с.

18. Уїлтон П., Колбі Дж. SQL для початківців: Пер. з англ. – М.: Видавничий дім «Вільямс», 2006. – 496 с.

19. Форту Б. Опануй самостійно SQL. 10 хвилин на урок. 3-є видання.: Пер. з англ. – М.: Видавничий дім «Вільямс», 2005. – 288 с.

20. Хернандес М., Вьескас Д. SQL-запити для простих смертних. Практичний посібник з маніпулювання даними в SQL. – М.: Изд-во «Лорі», 2003. – 459 с.

21. Хомоненко А.Д., Грідін В.В. Microsoft Access. Експрес-курс. – СПб.: БХВ-Петербург, 2005. – 304 с.

22. Чаповська Р., Жмеркевіч А. Робота з базами даних Microsoft Access 2000: Навчальний посібник. – Київ: Центр навчальної літератури, 2004. – 324 с.

23. Шевченко Н.А. Access 2003. Мистецтво створення бази даних. – М.: НТ Пресс, 2005. – 160 с.

24. Центр довідки з Access – Запити [Електроний ресурс]. – Режим доступу: <https://support.office.com/uk-ua/article/Queries-93fb69b7-cfc1-4f3e-ab56-b0a01523bb50>

25. Методичні вказівки до лабораторних робіт з предмету «Організація баз даних» / уклад. : В. В. Сидоренко, Л. В. Константинова; Кіровоградський національний технічний університет. – Кіровоград, 2013. – 64 с.

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторної роботи
за темою «Ознайомлення з основними командами мови SQL, що
забезпечують маніпулювання даними на прикладі СУБД Microsoft
Access»

для студентів спеціальностей

035 «Філологія»

121 «Інженерія програмного забезпечення»

122 «Комп'ютерні науки»

126 «Інформаційні системи та технології»

Укладачі:

ОРЛОВСЬКИЙ Дмитро Леонідович

БОРИСОВА Наталя Володимирівна

КОПП Андрій Михайлович

Відповідальний за випуск Годлевський М.Д.

Роботу до видання рекомендував Гамаюн І.П.

План 2020 р., поз. 270

Підп. до друку 29.09.2020. Гарнітура Times New Roman.

Ум. друк. арк. 2,8.

Видавничий центр НТУ «ХПІ»,

вул. Кирпичова, 2, м. Харків, 61002

Свідоцтво про державну реєстрацію ДК № 3478 від 21.08.2017 р.

Самостійне електронне видання